TRANSPERFECT
TRANSLATIONS

AFFIDAVIT OF ACCURACY

I, Michelle Ganswindt, hereby certify that the following is, to the best of my knowledge and belief, a true and accurate translation of the accompanying document [Q65540 Priority Document] from German into English.

ATLANTA
BOSTON
BRUSSELS
CHICAGO
DALLAS
DENVER
FRANKFURT
GENEVA
HONG KONG
HOUSTON
LONDON
LOS ANGELES
MIAMI
MINNEAPOLIS
MONTREAL
MUNICH
NEW YORK
PARIS
PHILADELPHIA
RESEARCH TRIANGLE PARK
SAN DIEGO
SAN FRANCISCO
SEATTLE
STOCKHOLM
TOKYO
WASHINGTON, DC

Michelle Ganswindt
TransPerfect Translations
601 13th Street, NW
Suite 320 North
Washington, DC 20005

Sworn to before me this
24th day of July, 2006

Signature, Notary Public

Lisa Sherfinski
Notary Public, District of Columbia
My Commission Expires 01-01-2008

Stamp, Notary Public

District of Columbia

Description

IP on CAN

1 The CAN Bus

The following description outlines only those CAN bus protocol characteristics that are absolutely necessary for further understanding. For a detailed description of the CAN bus reference is made to the literature.

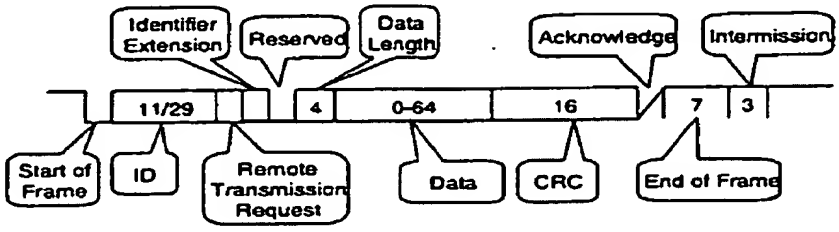

Figure 1: CAN frame format

Figure 1 depicts the CAN frame format. The CAN bus offers variable frame lengths with 0 to 8 octets of useful data, programmable gross transmission rates of up to 1 MBit/s (at 40 m segment length, the transmission rate is reduced at greater segment length due to fixed bit times) and message-based addressing. The latter means that a station is not assigned a network-wide address that serves as a destination address in a transmission process (e.g., MAC addresses in Ethernet). Instead, each message includes a content-related identifier in front, which other stations use to select messages that they are intended to receive. Since, from an application perspective, the messages can be viewed as CAN objects stored in the CAN hardware, received or to be transmitted, the identifiers used are also referred to as "CAN Object Identifiers" (COB IDs). They are either 11 bits or 29 bits long (extended CAN). Intelligent CAN controllers, which are standard today (e.g., the CAN interface of the 80C167 microcontroller or the Intel 82527 CAN controller), are capable of simultaneously configuring various transmit and receive IDs in their hardware buffers, the actual transmit or receive procedure of which is handled without involving a processor. Together with the excellent error detection and handling procedures, the CAN bus thus achieves an atomic multicast of the transmitted message.

The employed COB ID of a message simultaneously represents a priority: the lower the COB ID, the higher the priority of the message. The priority serves for bus arbitration, i.e., it is used to resolve access conflicts in case of simultaneous transmission requests by different nodes. In contrast to otherwise customary procedures, the conflict is resolved during transmission without the message being destroyed. The message with the highest priority is instead automatically put

through to the bus by using so-called dominant (logic 0) and recessive (logic 1) levels. For low priority messages that were set aside in this manner, the nodes independently attempt to re-transmit at the next possible instant.

Since a COB ID is normally assigned to the content-related meaning of the transmitted data (e.g., rotational speed front wheel left) and not to the transmitting or receiving node, two different nodes are prohibited from simultaneously transmitting a CAN packet with the same COB ID. Within the CAN protocol this is not permitted, since the two messages are superimposed and would result in a non-defined, or at least a faulty state on the bus.

2 Rough Architecture

The approach taken provides for the use of existing TCP/IP stacks (e.g., under PC/Windows NT) and makes the CAN bus appear as an Ethernet. To this end, Ethernet frames, which are typically transferred to an Ethernet driver, are transported in pieces via the CAN bus. A transmission protocol must therefore be developed, which transmits the Ethernet frames by means of suitable sequences of CAN packets between the stations involved. The information relevant to the transmission protocol to be developed must therefore be extracted from the Ethernet headers of the frames to be transmitted.

The resultant rough architecture is shown in Figure 2. The components to be developed consist in principle of a CAN driver layer to handle the CAN hardware and an Ethernet driver layer to communicate in downward direction with the CAN driver and interact in upward direction with the TCP/IP stack.

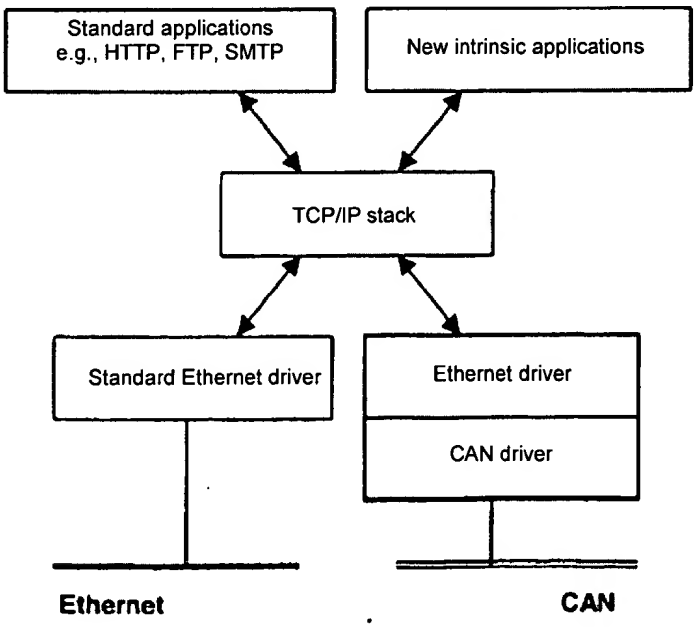

Figure 2: Rough architecture of the system to be developed

The depicted architecture has great advantages. If emulation of an Ethernet segment is completely successful, transmission via CAN is fully transparent for the TCP/IP stack and all application protocols based thereon. In particular, all the characteristics of the IP level that the stack usually provides, such as multicasting, routing, etc., are preserved. Seamless integration of the field level in superimposed networks is therefore given.

3 IP Packets and Ethernet Frames

This section describes the basic properties of IP packets as generated by superimposed protocols, such as TCP or UDP, and Ethernet frames to which the IP packets are typically mapped in LAN environments. For details, reference is made to the many textbooks on TCP/IP.

The data to be transmitted by the sending application on the transport level is usually transferred via the socket interface in the form of a TCP stream or UDP packet to the TCP/IP stack where it is disassembled or packaged into network level IP packets. Each IP packet is forwarded in the form of an Ethernet frame to the underlying link layer, while taking into consideration maximum lengths. As may be seen in Figure 4, the Ethernet header in particular contains the destination address of the frame on the Ethernet medium. It is termed the Media Access Control address (MAC address) and according to IEEE Standard 803.2 consists of a 48-bit identifier, which is unique for every Ethernet network adapter ever produced. The second field that must be looked at in the Ethernet header is the ether_type field in which the type of the packet contained in the Ethernet frame is defined. Here, the types ETHERTYPE_IP for IP packets and ETHERTYPE_ARP are being considered. The latter is used to map IP addresses to Ethernet MAC addresses by means of the Address Resolution Protocol (ARP), which is integrated in the TCP/IP stack.

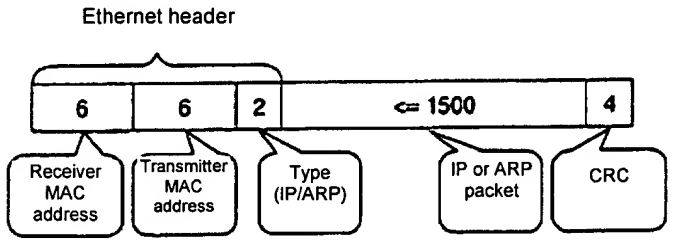

Figure 3: IP/ARP packet in the Ethernet frame

4 Problem of Addressing the Ethernet Layer

A basic problem is the assignment between the MAC addresses of the Ethernet frames and the CAN COB IDs, which are used by the CAN hardware for transmission. The following solution is proposed.

The fundamental difference between station-oriented addressing on the Ethernet side and message-oriented addressing within the CAN protocol has already been described in chapter 1. To realize transparent IP communication between the subscriber nodes thus requires unique addressing of the individual stations. In addition, the Ethernet standard provides for the possibility of broadcasts and multicasts, i.e., of Ethernet frames, which are simultaneously addressed to all or to a subset of the stations that are connected to the medium.

In contrast, CAN-based network nodes are generally capable of receiving any CAN packet transmitted on the bus if one of the receive buffers in the hardware is configured for the corresponding CAN identifier of the message. No unique assignment to a buffer needs to be made since the employed hardware also contains buffers whose acceptance of an identifier can be masked via a register so that the receive buffer will accept all COB IDs containing a certain bit pattern or even any CAN identifier. Likewise, each station is capable of transmitting CAN packets with different, freely selectable COB IDs by configuring corresponding send buffers.

A simple, naïve approach for mapping Ethernet addressing is to assign each subscriber a fixed COB ID. This cannot be a station-related receive address, however, since this would violate the prohibition of simultaneously putting two CAN messages with the same ID on the bus if the respective station is to receive messages from two different nodes at the same time. This would leave the assignment of the transmitter identifier as the ID uniquely assigned to a node. The station addressed as the receiver, however, could then be encoded only in the data portion of the CAN message. A station would consequently have to receive all of the transmitted messages, evaluate their content and discard the packets addressed to other stations in order to determine whether it was addressed directly. This would not only increase the system load to the maximum, but the address of the target station would in addition have to be accommodated in the useful data of each packet, which is already tight at 8 bytes.

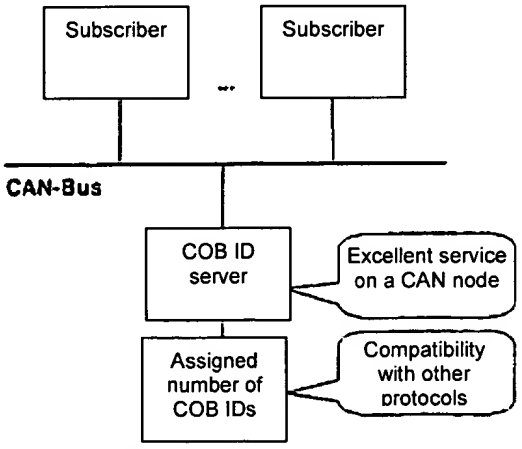

Figure 4: Management of CAN object identifier by COB ID server

Here, a model was developed (see Figure 5) which avoids the aforementioned problems but requires a node in a CAN segment, which acts as a central instance. This node is the so-called

COB ID server, which manages the number of usable COB IDs. The number of COB IDs that can be used for Ethernet emulation can be freely defined. This ensures the required compatibility with the other CAN protocols on the CAN bus. For the transport of the useful data, which forms Ethernet frames, each pair of nodes communicating with one another is associated with a corresponding pair of COB IDs. For broadcasting, each node ready for sending receives a COB ID from the COB ID server. The receivers involved follow the assignment of COB IDs and prepare to receive packets with these identifiers and therefore ignore packets intended for other stations.

The assignments that have been made can be readily maintained for subsequent communication processes between node pairs. Only when free COB IDs become increasingly scarce does the COB ID server have to request the assigned COB IDs to be returned. The protocol for managing the COB IDs will be described in detail in the following chapter 5.

For correct initialization, each subscriber node must nevertheless have a unique station address within the Ethernet emulation of a CAN segment, which is already defined at the time of installation. This address serves to identify the node within the initialization phase of the protocol between the node and the COB ID server. For this purpose, an unsigned 16-bit value was selected, so that a theoretical maximum of 64k nodes can be addressed. At the same time, this value forms the lower 16 bits of the Ethernet MAC address of the station in the context of the emulation. The higher value bits are set on a fixed prefix, e.g., 0 (cf. Figure 6). Since this is only an Ethernet emulation and not the actual medium, these MAC addresses cannot conflict with those of "genuine" Ethernet adapters on a segment.

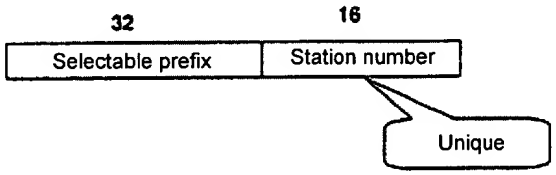

Figure 5: MAC addresses of CAN nodes

5 The COB ID Server Protocol

The Ethernet emulation layer integrated into the subscriber nodes and the application of the COB ID server together implement a protocol which permits the assignment of COB IDs for different purposes and regulates flow control in the communication sequence between the nodes. The Ethernet layer of the subscriber nodes acts as a protocol layer between TCP/IP implementation and the CAN driver (cf. Figure 2). Its job is to map the Ethernet MAC addresses to COB IDs, segment and reassemble Ethernet frames in CAN packets, and manage the associated CAN objects in the hardware with the aid of the CAN driver.

The job of the COB ID server (hereinafter also referred to as "server" if unambiguous) is to store and organize the management information for handling the protocol between the subscriber nodes, particularly for the correct and efficient use of COB IDs to transmit Ethernet frames.

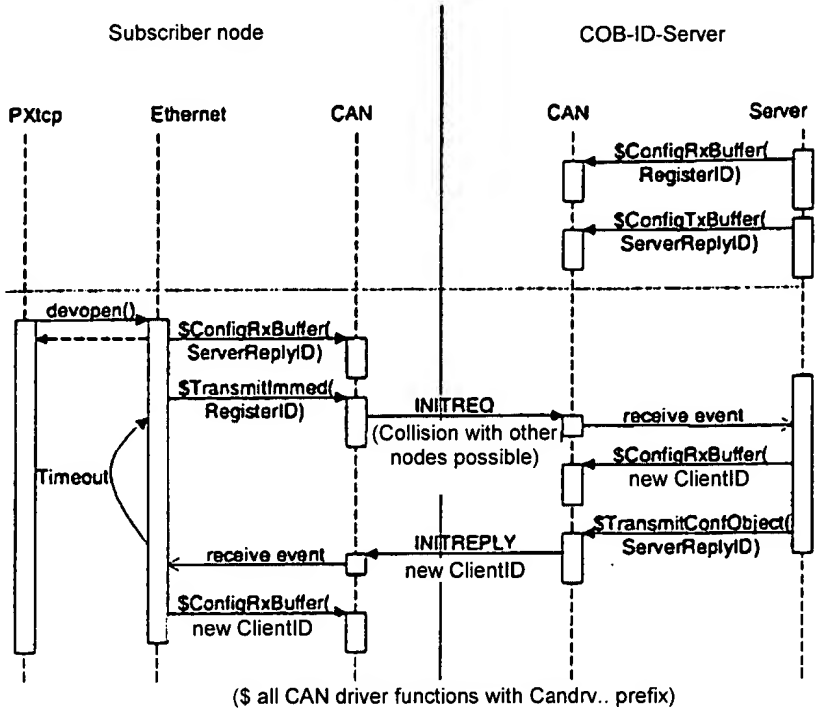

Figure 6: Registration of a user in the COB ID server

The COB ID server protocol includes a dynamic initialization protocol, which a new subscriber uses to register with the COB ID server. The sequence is illustrated in Fig. 7.

As soon as a subscriber node has locally initialized itself, it sends a registration request with a well known COB ID to the COB ID server. This well known COB ID is determined at the time of configuration and is used for registration by all subscribers. To this extent, a collision may theoretically occur, namely if several subscribers simultaneously request registration, which would lead to a detected faulty transmission on the CAN bus. This conflict is resolved by using the CSMA/CD mechanism.

If registration is successful, the server assigns a private unique COB ID to the registering subscriber, which the subscriber station then uses for all further communication with the server. In this response message and in any further control messages, the server uses a second well known COB ID, for which all nodes participating in the protocol must be continuously ready to receive status messages or control messages. This also implies, however, that all nodes will receive every message of the server (multicasting). Each of the receivers must use the corresponding packet content to evaluate whether the message concerns it. Since multicasting of control messages occurs only rarely compared to useful data transmissions, the complexity thereby generated must be regarded as low.

The two well known COB IDs discussed above are the only two identifiers reserved by the "IP via CAN" protocol that are defined throughout the system at the time of installation. All other COB IDs are assigned by the server and may be withdrawn after use if necessary. For the COB IDs that are available in this connection, fixed areas are currently defined in the server code. A dynamic definition of corresponding COB ID pools, e.g., at the time of initialization of the server, can especially take into account the needs of other applications that coexist on the CAN bus. Thus, the requirement for compatibility of the protocol with other CAN protocols is optimally met.

Transmission orders of the higher protocol layer for Ethernet frames and the resultant procedures in the COB ID server protocol will now be discussed. If, in a subscriber node, an Ethernet frame is forwarded for transmission the destination address in the Ethernet header is first evaluated and the further procedure will depend on whether it is a real station address or a broadcast address. (A multicast address, which is also possible, is treated like a broadcast transmission and will not be further discussed here.)

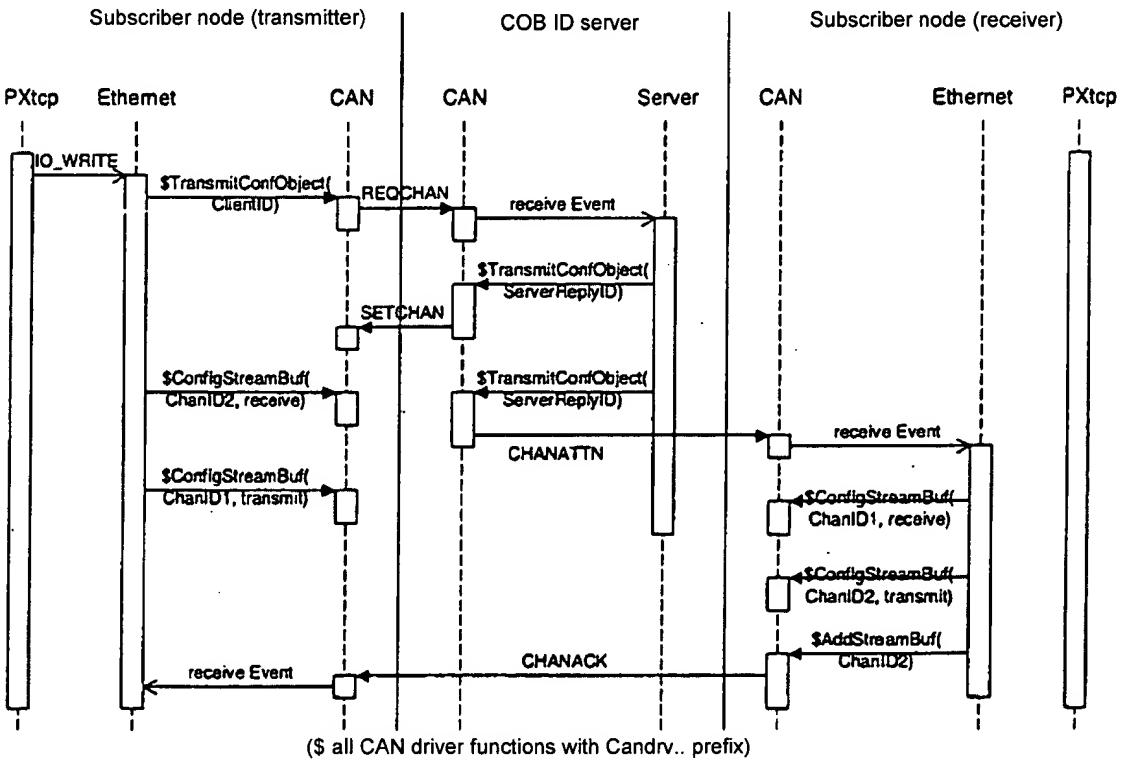

Figure 7: Assignment of a point-to-point COB ID

A point-to-point transmission of an Ethernet frame to a specific station with a given MAC address starts with the request of a COB ID pair from the COB ID server as shown in Figure 8. The COB ID to be used is assigned, as described above, via the well known COB ID. It is also registered by all the other stations on the bus, but is taken into account only by the partner station

of the sender, which gets ready to receive the first identifier and then sends a CAN packet, addressed with the second assigned COB ID as an acknowledgment to the first station, which has prepared to receive this identifier. The Ethernet frames can now be transmitted.

The sending node starts transmission of the Ethernet frames (cf. Figure 9) with a first CAN packet that contains the length of the Ethernet frame and the first useful data. All other nodes receive this and the immediately following CAN packets without any additional control information and put them together to form a received Ethernet frame. When transmission of the frame is complete, the frame is forwarded to the upper network layers.

If this is a broadcast packet, it is handled similarly to the point-to-point transmission. The transmitting subscriber node requests a broadcast COB ID from the COB ID server. The COB ID to be used is again assigned via the well known COB ID and is registered by all other stations on the bus, which in this case can configure themselves with the new COB ID to receive the broadcast frame.

The COB IDs that are assigned for a specific purpose can be reused in a renewed communication between the same station pair until the COB ID server withdraws them due to a lack of available COB IDs. In a point-to-point transmission in the opposite direction to a previous transmission, the assigned COB ID pair is used again; the participating stations alone are decisive. The same applies to renewed broadcasts.

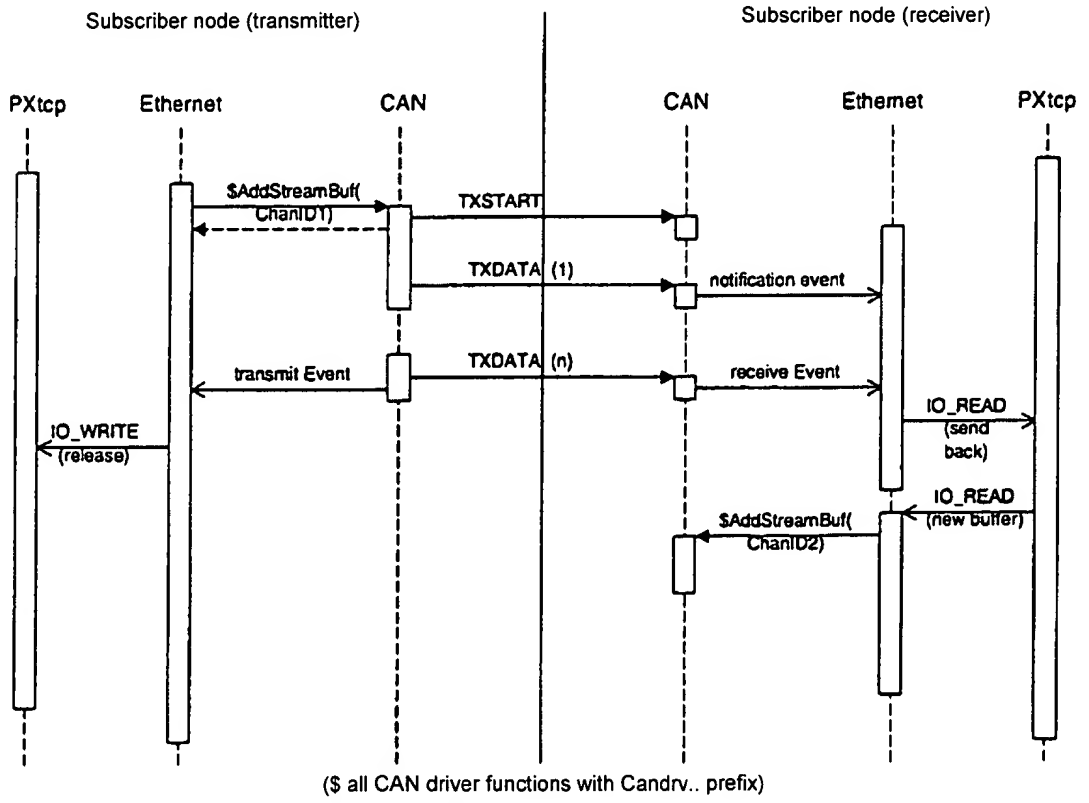

Figure 8: Transmission of an Ethernet frame

The recall of a COB ID or pair assigned by the COB ID server is initiated when the number of COB IDs still available for assignment in the COB ID server falls below a critical value. The sequence and scope of the recall can be implemented as a function of a strategy. A simple implementation that has been used until now employs a FIFO algorithm. The COB ID server can also monitor the actual usage of assigned COB IDs by the subscriber nodes over time and use this information as the basis, for example, for an LRU algorithm for selecting the recalls. To initiate a recall, the COB ID server sends to each station involved a return request that contains the station address of the node and the corresponding COB ID. Any packet that is still being transmitted over the corresponding channel can be completed by the node before the return of the COB ID is confirmed by the subsequently transmitting node by means of a return response to the COB ID server. Finally, the COB ID server adds the returned COB IDs again to its supply of available COB IDs and acknowledges the return with a corresponding control message. The nodes can then deinitialize the associated send and receive objects and remove the entries from the local assignment tables. The acknowledgment by the COB ID server is provided to prevent a node from losing its receiving readiness for a COB ID before the distant terminal has adjusted all transmission procedures to this ID. The described process for a COB ID pair applies correspondingly to COB IDs assigned for broadcast purposes, in which case only one node and the COB ID server are involved.